Cálculo Numérico

Um guia prático com Python

Prof. Dr. Rogério Vargas[1]

[1]Centro de Estudos do Mar
Universidade Federal do Paraná

2024

GEOIDELAB

UFPR

Vamos nos conhecer?

Olá! Bem-vindo a componente **Cálculo Numérico**. Este material contém um direcionamento das aulas realizadas.

O professor
Apresentação no prezi. Clique aqui.

Aprovação
Frequência e avaliações.

Mas afinal, o que veremos na componente?

Python

Iremos implementar várias funções utilizando a linguagem de programação Python a partir da versão 3.

- Python.org (oficial)
  - Google Colab
  - OnlineBGD
  - Replit

Conteúdo das Aulas I

- Python

1. Por que estudar Cálculo Numérico?
   - Exemplos
   - Objetivos
   - Grave isso
   - Reprodução da solução babilônica
2. Ambiente de programação Google Colab
3. Aritmética de Ponto Flutuante
4. Aritmética de Ponto Flutuante + Aplicações
5. Zero Funções
6. Zero Funções
7. Funções em Python
8. Sistemas Lineares
9. Sistemas Lineares + Aplicações

Conteúdo das Aulas II

Encontro 3: Por que estudar Cálculo Numérico?

Exemplos

- Foguete Ariane 5 (veja o vídeo)
  - Ano 1996
  - US$ 500 mi (foguete)
  - US$ 7 bi (projeto)
  - Over ow foi a causa
- Código em Python

```
1 # Soma de pontos
    flutuantes
2 a = 0.1
3 b = 0.2
4 c = a + b
5 print(c)
6 # Output:
    0.30000000000000004
```

Objetivos

- Reconhecer a importância do cálculo numérico
- Conhecer princípios básicos usados em cálculo numérico.
- Reconhecer problemas que podem ser resolvidos por cálculo numérico.
- Estabelecer fases para a resolução de problemas reais.

Objetivos

Compreender como os números são representados nas calculadoras e computadores e como são realizadas as operações numéricas nestes sistemas digitais.

Entender o que são métodos numéricos de aproximação, como e por que utilizá-los.

Identi car problemas que requerem o uso de técnicas numéricas para a obtenção de sua solução.

Objetivos

- Conhecer e aplicar os principais métodos numéricos para a solução de problemas clássicos. Ex.: obter zeros reais de funções reais, resolver sistemas de equações lineares, fazer interpolação polinomial, ajustar curvas e fazer integração numérica
- Estimar e analisar os erros obtidos devido à aplicação de métodos numéricos e propor soluções para minimizá-los e se possível, eliminá-los

Grave isso

Em um método numérico, uma solução aproximada é obtida de forma construtiva:

1. Partindo de aproximações iniciais, vão sendo construídas novas aproximações até que uma aproximação considerada "boa seja obtida.
2. Um método numérico pode ser escrito em forma de algoritmo com as operações (ou grupos de operações), podendo ser executadas repetidamente.

Reprodução da solução babilônica

Passo 1: Construção dos triângulos

> Desenhar 3 triângulos ($T1$, $T2$ e $T3$), todos retângulos (um ângulo reto) e isóceles (dois catetos iguais) de tamanhos diferentes.

Sugestão: Construir os triângulos com catetos iguais 10 cm, 20 cm e 30 cm.

Reprodução da solução babilônica

Passo 2: Medida da Hipotenusa

Marcar na hipotenusa (H) o tamanho de um cateto inteiro.

$$H = 1(\text{cateto})$$

Observações:

1. Independentemente do triângulo, só será possível marcar um único cateto.
2. Observe que que a medida $H = 1$ é pouco precisa.

Reprodução da solução babilônica

Passo 3: Melhorando a precisão da medida

Dividir o cateto em 10 partes iguais e complementar o comprimento da hipotenusa com quantos décimos de catetos inteiros for possível.

H = 1 (cateto) + 4 (catetos/10) + :::

Reprodução da solução babilônica

Pausa para re exão

Chamando o cateto de C e o tamanho da hipotenusa de H temos:

$$H = 1 \quad C + 4 \quad \frac{C}{10} + ? \quad C\frac{C}{100} + ?? \quad \frac{C}{1000} + :::$$

Quantas vezes conseguimos dividir o cateto por 10?

Existe uma barreira física (tamanho do triângulo) que limita essas divisões.

Mas essa barreira existiria se fosse feito por um algoritmo?

Cálculo numérico

Considerações nais

As soluções numéricas dependem da qualidade do modelo matemático, assim como do próprio método utilizado para a solução.

Existem vários métodos que resolvem o mesmo problema, estudaremos alguns deles para comparar a qualidade, defeitos e aplicabilidade.

Encontro 4: Ambiente de programação Google Colab

Google Colab

- Conta Google
  - Criar conta e/ou;
  - Logar na conta;
  - Acessar o Google Colab;
  - Seguir instruções da aula;
  - Pratique.

Encontro 5: Aritmética de Ponto Flutuante

Objetivos

Conversões entre bases.

Notação cientí ca.

Arredondamento e erros.

Aritmética de ponto utuante.

Compreender os erros nas conversões.

Compreender o código em Python de erro de arredondamento.

Conversão de decimal para binário

Conversão de binário para decimal

Conversão de bases

Converta os seguintes valores da sua base para binário ou decimal

1. $x = 2_{(10)}$
2. $x = 123_{(10)}$
3. $x = 10110_{(2)}$
4. $x = 1111_{(2)}$
5. $x = 1111_{(10)}$

Notação cientí ca

Notação cientí ca é uma forma de expressar números muito grandes ou muito pequenos de maneira mais simpli cada.

> De nição:
> Formalmente, um número está em notação cientí ca quando é representado como o produto de um número real (a), tal que (1 j aj < 10), e uma potência de 10, escrita como (10^n), onde (n) é um número inteiro. A forma geral é:
>
> a 10^n

onde a é a mantissa e n o expoente. Sendo,1 a < 10 ou 10 < a 1 e n um número inteiro.

Notação cientí ca

Converta de notação cientí ca para decimal:

1. $x = 1\,;2 \;\; 10^{5}$
2. $x = 9\,;2 \;\; 10 \;\; ^{5}$
3. $x = \;\; 7;2123 \;\; 10^{7}$

Notação cientí ca

Converta de decimal para notação cientí ca:

1. 1250000
2. 500000000000
3. 0; 0000256
4. 0; 0000003
5. 0; 0100003

Representação de um número

A representação de um número pelo computador é dada por:

$$0{:}d_1 d_2 d_3 {:}{:}{:} d_t \quad {}^{n}$$

onde d_1 até d_t é a mantissa, é base e n é o expoente. Sendo:

d_1 ≠ 0

n 2 [m; M] no qual m é o limite inferior e M é o limite superior do expoente.

Conversão de um número

Representação

$$x = 0{:}d_1 d_2 d_3 :: d_t \quad {}^{n}$$

1. $x = 2_{(10)}$
2. $x = 235;89_{(10)}$
3. $x = 0;000875_{(10)}$

Aritmética de ponto utuante

Operação

$$F(B; t; m; M\ )$$

onde:

- B é a base.
- t é o número de dígitos.
- m é limite inferior do expoente.
- M é o limite superior do expoente.

Arredondamento e erros

De nição: Arredondamento por Truncamento
Se desejamos arrendondar utilizando DIGSE= 3, devemos simplesmente ignorar os dígitos restantes após o terceiro signi cativo.

a saber que DIGSE signi ca DIGitos Signi cativos Exatos.

Arredondamento e erros

De nição: Arredondamento por aproximação

Se desejamos arrendondar utilizando DIGSE= 3, devemos olhar para o dígito seguinte a esse, ou seja, o quarto, o qual usaremos a notação d_4. Se esse dígito for entre 0 e 4 não mudamos o terceiro dígito d_3, porém se for entre 5 e 9 adicionamos uma unidade ao terceiro dígito.

Arredondamento e erros

Exemplo

Represente os números $x_1 = 0;437$, $x_2 = 0;144$, $x_3 = 0;495$ e $x_4 = 0;31415926510^1$ com dois dígitos signi cativos por truncamento e arredondamento.

Arredondamento e erros

Under ow

Ocorre quando o resultado de uma operação de ponto utuante é menor em magnitude (ou seja, mais próximo de zero) do que o menor valor representável como um número de ponto utuante normal no tipo de dados de destino. Isso signi ca que o resultado é tão pequeno que não pode ser adequadamente representado na memória do computador.

$$[x < \text{fmin}_N]$$

onde:

- x é o resultado da operação.
- (fmin_N) é o menor valor positivo normal representável em ponto utuante.

É importante considerar essas limitações ao trabalhar com cálculos numéricos em ponto utuante.

Arredondamento e erros

Over ow

Ocorre quando o resultado de uma operação de ponto utuante é maior em magnitude do que o maior valor representável como um número de ponto utuante normal no tipo de dados de destino. Isso signi ca que o resultado é tão grande que não pode ser adequadamente representado na memória do computador.

$$x > \text{fmax}_N]$$

onde:

- x é o resultado da operação.
- (fmax_N) é o maior valor positivo normal representável em ponto utuante.

É importante considerar essas limitações ao trabalhar com cálculos numéricos em ponto utuante.

Código-fonte de erro de arredondmento

O código-fonte em Python mostra um exemplo de erro ocasionado por aritmética de ponto utuante.

Encontro 6: Aritmética de Ponto Flutuante + Aplicações

Tipos de erros

Vamos considerar basicamente dois tipos de erros: Absoluto e Relativo. Estas de nições nos auxiliam no estudo da tolerância na aplicação de algum método numérico, bem como na sua convergência, que veremos mais adiante.

1. Absoluto
2. Relativo

Tipos de erros

De nição:
Erro absoluto EA ,

$$EA(x) = |x - \tilde{x}|$$

onde x tomamos como o valor exato e $\tilde{x}$ a aproximação para o mesmo.

Tipos de erros

De nição:
Erro relativo ER,

$$ER(x) = \frac{jx \quad xj}{jxj} = \frac{EA}{jxj}$$

Nota-se que o erro relativo é uma medida adimensional. Este conceito torna o erro relativo uma proporção com relação ao valor real.

Tipos de erros

Exemplo

Sejam $x = 123456{,}789$ e sua aproximação $\bar{x} = 123000$. O erro absoluto é

$$|x - \bar{x}| = |123456{,}789 - 123000| = 456{,}789$$

e o erro relativo é

$$\frac{|x - \bar{x}|}{|x|} = \frac{|456{,}789|}{|123456{,}789|} = 0{,}00369999$$

ou ainda, 0,36%.

Aplicações

Converta para binário:

- 11
- 10
- 32
- 65
- 127
- 200
- 1401
- 4095

Converta para o sistema decimal:

- 10_2
- 11_2
- 111_2
- 10101_2
- 11110_2
- 11111_2
- 110010111_2
- 1000000001_2

Aplicações

Converta para notação cientí ca:

- 15
- 123
- 325678
- -457
- 0,765
- 0,00000012
- -0,999900
- -893,999812

Converta para o sistema decimal:

- $1;5\ \ 10^{1}$
- $28\ \ 10^{6}$
- $3;982\ \ 10^{1}$
- $9;82\ \ 10^{3}$
- $1;982\ \ 10^{8}$
- $9;2\ \ 10^{2}$
- $7;8254\ \ 10^{7}$
- $3;72319832\ 10^{4}$

Aplicações

Arredondamento:

0; 1234 10 (DIGSE=3)

0; 888888 10 (DIGSE=4)

0; 3413 10 (DIGSE=2)

0; 487 10 (DIGSE=1)

0; 4231 10 (DIGSE=3)

0; 1234674 10 (DIGSE=2)

0; 12345 10 (DIGSE=4)

0; 1234555552 10 (DIGSE=6)

Truncamento:

0; 1234 10 (DIGSE=3)

0; 888888 10 (DIGSE=4)

0; 3413 10 (DIGSE=2)

0; 487 10 (DIGSE=1)

0; 4231 10 (DIGSE=3)

0; 1234674 10 (DIGSE=2)

0; 12345 10 (DIGSE=4)

0; 1234555552 10 (DIGSE=6)

Aplicações

Calcule o erro relativo e o erro absoluto de uma área de um círculo, sabendo:

1. R = 100m
2. $_1 = 3;14$
3. $_2 = 3;141592$
4. A = R^2

Aplicações

Utilizando o Python e uma uma ferramenta de IA, faça os seguintes programas:

1. Exiba o maior número de casas decimais do .
2. Faça um programa em Python, onde o usuário informe um número um número (inteiro) e o programa converta se for decimal para binário e vice-versa.
3. Faça um programa em Python que o usuário informe o número decimal e o programa exiba em notação cientí ca. Faça também o processo inverso.

Encontro 7: Zero Funções

Objetivos

O que é

Funções de 1^{o}, 2^{o} e 3^{o} grau.

Métodos numéricos para determinação de zero funções

Processo iterativo

Grá co em Python

Método de Bolzano

Método da bissecção

Critério de parada

Exemplo no Excel

Zero Funções

> O que é?
> É onde a curva do grá co intercepta o eixo x e $y = 0$. No ponto que é interceptado há uma raiz.

Funções de:

- Primeiro grau
- Segundo grau
- Terceiro grau

Função de primeiro grau

As funções do tipo $y = ax + b$ ou $f(x) = ax + b$, onde a e b assumem valores reais e $a \neq 0$ são consideradas funções do 1º grau.

A função é crescente quando $a > 0$ e;

Decrescente quando $a < 0$.

Raiz de uma função de primeiro grau

Raiz é o valor de x que anula $f(x)$. Para calcular a função é necessário igualar a equação a zero.

$f(x) = y = 0$

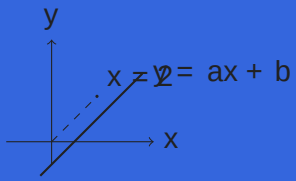
$f(raiz) = y = 0$

$f(x) = ax + b$

$ax + b = 0$

$ax = b$

$x = \frac{b}{a}$

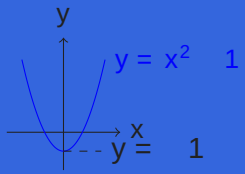


Raiz de uma função de segundo grau

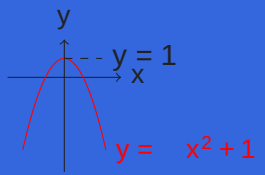
Raiz é o valor de x que anula $f(x)$. Para calcular a função é necessário igualar a equação a zero. Utilizando a fórmula de Bhaskara é possível encontras as raízes da função x^0 e x^{00}.

Além disso, o valor do delta nos permite saber quantos zeros a função quadrática possuirá.

- > 0: duas raízes reais distintas;
- $= 0$: uma única raiz real;
- < 0: não possui raiz real.



(a) Parábola com concavidade para cima.



(b) Parábola com concavidade para baixo.

Figure 1: Grá cos de parábolas com concavidade para cima e para baixo

Exemplos

Pode-se observar os seguintes exemplos:

1. $f(x) = x \quad 3$
2. $f(x) = \frac{8}{3}x \quad 4$
3. $f(x) = x^2 \quad 5x + 6$

Exemplos

Porém, nem sempre é possível encontrar analiticamente a raiz de uma função, por exemplo:

$$f(x) = x^3 + 2x^2 \quad x + 1$$

$$\sin(x) + e^x$$

$$x + \ln(x)$$

Métodos numéricos para determinação de zeros de funções

A ideia central destes métodos é partir de uma aproximação inicial para a raiz e em seguida re nar essa aproximação através de um processo iterativo .

1. Isolar cada zero que se deseja determinar da função f em um intervalo [a; b], sendo que cada intervalo deverá conter um e somente um zero da função f .
2. Calcular a raiz aproximada através de um processo iterativo até a precisão desejada.

Processo iterativo

Iteração
É o ato de iterar (repetir) uma função por um determinado período de tempo até que uma condição seja alcançada.

Há uma ampla variedade de métodos numéricos que consistem em processos iterativos. Esses processos são de nidos pela repetição de uma operação especí ca.

A essência desse tipo de procedimento é realizar um cálculo especí co repetidamente, de modo a obter, a cada iteração subsequente, um resultado mais re nado em comparação com o da iteração anterior.

É fundamental ressaltar que em cada iteração, o resultado obtido na iteração anterior é utilizado como parâmetro de entrada para o cálculo subsequente.

Processo iterativo

Há vários aspectos comuns a todos os processos iterativos:

Estimativa inicial

Para iniciar um processo iterativo, é necessário possuir uma estimativa inicial do resultado do problema. Esta estimativa pode ser adquirida de várias maneiras, dependendo da natureza do problema.

Convergência

Para alcançar um resultado próximo do esperado, é crucial que a cada passo ou iteração, nosso resultado se aproxime cada vez mais do valor desejado.

Critério de parada

Não é viável continuar um processo numérico inde nidamente. É necessário interrompê-lo em algum momento. Determinar quando parar as iterações de um processo numérico é chamado de critério de parada, o qual varia conforme o problema em questão e a precisão desejada para obter a solução.

Processos iterativos

Há diversos métodos para encontrar os zeros de uma função:

1. Método da Bissecção;
2. Método do Ponto Fixo (MPF);
3. Método de Newton-Raphson;
4. Método da Secante;
5. Método de Muller;
6. Método de Brent.

De nição do Trabalho 1.

Trabalho 1

1. De nir o método numérico;
2. Tempo de apresentação de até 1 hora;
3. Ter fundamentação teórica;
4. 1 ou 2 exemplos;
5. Planilha com as iterações;
6. Função em Python com o método (deverá ser implementado sem o uso de nenhum biblioteca/funções exceto as matemáticas);
7. Lista de 4 exercícios resolvidos;
8. Elaborar a apresentação (preferência em LaTeX);
9. Mostrar prévia do trabalho 1 semana antes da entrega;
10. Avaliação individual, todos devem apresentar.

Função de terceiro grau

Gerar grá co em Python

Método de Bolzano

Teorema

Seja uma função $f(x)$ contínua em um intervalo $[a; b]$, tal que, $f(a)\ \ f(b) < 0$, então a função $f(x)$ possui pelo menos uma raiz no intervalo $[a; b]$.

O teorema assegura que se f troca de sinal nos pontos a e b então f tem pelo menos um zero entre estes pontos .

Exemplo

Seja a função f (x) = x ln (x) 3; 2.
Calcule o valor de f (x) para valores arbitrários de x, como mostrado na tabela abaixo:

x	1	2	3	4
f (x)	-3,20	-1,81	0,10	2,36

Pelo teorema de Bolzano, concluímos que existe pelo menos uma raiz real no intervalo [2; 3].

Método da bissecção

O processo consiste em dividir o intervalo que contém o zero ao meio e por aplicação do Teorema de Bolzano, aplicado aos subintervalos resultantes, determinar qual deles contém o zero.

$$a; \frac{a+b}{2} \quad ; \quad \frac{a+b}{2}; b$$

O processo é repetido para o novo subintervalo até que se obtenha uma precisão pre xada. Desta forma, em cada iteração o zero da função é aproximado pelo ponto médio de cada subintervalo que a contém.

Método da bissecção: Critério de parada

O critério de parada é frequentemente baseado no valor de .

é a tolerância desejada para a precisão da raiz.

O processo iterativo continua até que o intervalo de busca seja su cientemente pequeno, ou seja, jb aj < .

Quando a largura do intervalo é menor do que , a raiz é considerada su cientemente precisa.

jb aj <

Encontro 8: Zero Funções + Aplicações

Aplicações

O objetivo desta aula é resolver problemas práticos com programação.

1. Abra a página rogerio.in ;
2. Vá na componente Cálculo Numérico;
3. Siga as instruções do professor.

Encontro 9: Funções em Python

Aplicações

O objetivo desta aula é aprender/recordar o uso de funções em Python.

1. Siga as instruções do professor.

Encontro 10: Sistemas Lineares

Objetivos

De nição

Introdução

Solução do sistema

Método de Gauss

Sistema Lineares

Sistemas lineares são um conjunto de equações lineares que deverão ser resolvidas ao mesmo tempo.

Equação linear

Uma equação linear nas incógnitas $x_1; x_2; :::; x_n$ é uma equação que pode ser expressa na forma padrão

$$a_1x_1 + a_2x_2 + ::: + a_nx_n = c$$

em que $a_1; a_2; :::; a_n$ é chamado de coe cientes . O $x_1; x_2; :::; x_i$ são as incógnitas e por m, c é chamado de termo independente.

Sistemas Lineares

Se $c = 0$! Equação linear homogênea

Exemplo:

$$x + 2y \quad 4z = 7$$
$$x \quad 5z = 15$$
$$t + 2u + 3v = 0$$

Sistemas Lineares

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \quad + a_{1n}x_n = c_1 \\ a_{21}x_1 + a_{22}x_2 + \quad + a_{2n}x_n = c_2 \\ a_{31}x_1 + a_{32}x_2 + \quad + a_{3n}x_n = c_3 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \quad + a_{mn}x_n = c_m \end{cases}$$

Sistema linear com:

m equações

n incógnitas

Exemplos:

$$\begin{cases} 3x + 2y = 5 \\ x + 3y = 4 \end{cases}$$

$$\begin{cases} 4x + 3y + 5w + z = 2 \\ 5x + 3y + w + 2z = 1 \\ 3x + 5y = 5 \end{cases}$$

Sistemas Lineares

Solução do sistema

$$(x_1; x_2; x3; \quad ; x_n)$$

Exemplo:

$$\begin{cases} x + y = 3 \\ x \quad y = 1 \end{cases}$$

Sistema linear homogêneo:
Exemplo:

$$\begin{cases} 5x \quad y \quad 4z = 0 \\ 2x + y \quad 3z = 0 \\ x + 2y \quad 3z = 0 \end{cases}$$

Sistemas Lineares

Exemplo:

$$\begin{cases} x - y - z = 0 \\ x - 2y - 2z = 0 \\ 2x + y + z = 0 \end{cases}$$

Vamos validar com:

$(0; 0; 0)$

$(0; 1; \quad 1)$

$(1; 1; 1)$

Sistemas Lineares

Método da Eliminação de Gauss

Processo

Consiste em transformar o sistema $Ax = b$ em um sistema equivalente com matriz dos coe cientes triangular superior, por meio de transformações elementares ou transformações equivalentes.

$$Ax = b \quad \Rightarrow \quad \tilde{A}x = \tilde{b}$$

Sistemas Lineares

Método da Eliminação de Gauss

Teorema

Seja $Ax = b$ um sistema linear, se:

1. Trocarmos duas equações;
2. Multiplicarmos uma equação por uma constante não-nula;
3. Adicionarmos um múltiplo de uma equação a outra equação.

O método da eliminação de Gauss engloba duas fases:

1. Fase de eliminação;
2. Fase de substituição.

Sistemas Lineares

1. Fase de eliminação

1. Transformações elementares na matriz aumenta [Ajb]
2. Para uma matriz (n n) este processo terá (n 1) etapas

Procedimento

1. Montar a matriz aumenta [Ajb]
2. Determinação do pivô: a_{kk}
3. De nir os multiplicadores de linha

$$m_{ik} = \frac{a_{ik}}{a_{kk}}$$

4. Atualização das linhas

$$L_i \quad L_i \quad m_{ik} \quad L_{pivô}$$

Sistemas Lineares

1. Fase de eliminação

1. Transformações elementares na matriz aumenta $[Ajb]$
2. Para uma matriz (n n) este processo terá (n 1) etapas

Procedimento

1. Montar a matriz aumenta $[Ajb]$
2. Determinação do pivô: a_{kk}
3. Definir os multiplicadores de linha

$$m_{ik} = \frac{a_{ik}}{a_{kk}}$$

4. Atualização das linhas

$$L_i \quad L_i \quad m_{ik} \quad L_{\text{pivô}}$$

Sistemas Lineares

Exemplo

$$\begin{cases} x_1 + x_2 + 3x_4 = 4 \\ 2x_1 + x_2 + x_3 + x_4 = 1 \\ 3x_1 + x_2 + x_3 + 2x_4 = 3 \\ x_1 + 2x_2 + 3x_3 + x_4 = 4 \end{cases}$$

$$A^{(0)}|b^{(0)} = \left[\begin{array}{cccc|c} 1 & 1 & 0 & 3 & 4 \\ 2 & 1 & 1 & 1 & 1 \\ 3 & 1 & 1 & 2 & 3 \\ 1 & 2 & 3 & 1 & 4 \end{array}\right]$$

Sistemas Lineares

Passo 1: Quem é o pivô? Tudo abaixo do pivô deverá ser zero.

Pivô

$$\left[\begin{array}{cccc|c} 2 & 1 & 0 & 3 & 4 \\ 2 & 1 & 1 & 1 & 1 \\ 3 & 1 & 1 & 2 & 3 \\ 1 & 2 & 3 & 1 & 4 \end{array}\right]$$

Sistemas Lineares

Passo 2: Operar nas linhas

$L_2 \quad L_2 \quad 2L_1$

$$\left[\begin{array}{cccc|c} 1 & 1 & 0 & 3 & 4 \\ 0 & 1 & 1 & 5 & 7 \\ 3 & 1 & 1 & 2 & 3 \\ 1 & 2 & 3 & 1 & 4 \end{array}\right]$$

Sistemas Lineares

Passo 2: Operar nas linhas

$L_3 \quad L_3 \quad 3L_1$

$$\left[\begin{array}{cccc|c} 1 & 1 & 0 & 3 & 4 \\ 0 & 1 & 1 & 5 & 7 \\ 0 & 4 & 1 & 7 & 15 \\ 1 & 2 & 3 & 1 & 4 \end{array}\right]$$

Sistemas Lineares

Passo 2: Operar nas linhas

$L_4 \quad L_4 + L_1$

$$\left[\begin{array}{cccc|c} 2 & 1 & 0 & 3 & 4 \\ 6 & 6 & 6 & 6 & 7 \\ 0 & 1 & 1 & 5 & 7 \\ 6 & 6 & 6 & 6 & 7 \\ 0 & 4 & 1 & 7 & 15 \\ 6 & 6 & 6 & 6 & 5 \\ 0 & 3 & 3 & 2 & 8 \end{array}\right]$$

Concluído para a primeira coluna. Repetir agora para a segunda coluna. Trocar o pivô.

Sistemas Lineares

Passo 2: Operar nas linhas

$L_3 \quad L_3 \quad 4L_2$

$$\left[\begin{array}{cccc|c} 2 & 1 & 0 & 3 & 4 \\ 0 & 1 & 1 & 5 & 7 \\ 0 & 0 & 3 & 13 & 13 \\ 0 & 3 & 3 & 2 & 8 \end{array}\right]$$

Sistemas Lineares

Passo 2: Operar nas linhas

$L_4 \quad L_4 + 3L_2$

$$\left[\begin{array}{cccc|c} 2 & 1 & 0 & 3 & 4 \\ 0 & 1 & 1 & 5 & 7 \\ 0 & 0 & 3 & 13 & 13 \\ 0 & 0 & 0 & 13 & 13 \end{array}\right]$$

Concluído, valores abaixo da diagonal são zeros.

Sistemas Lineares

Passo 3: Reescrever o sistema e resolver

Sistema reescrito

$$\begin{cases} x_1 + x_2 + 3x_4 = 4 \\ x_2 + x_3 + 5x_4 = 7 \\ 3x_3 + 13x_4 = 13 \\ 13x_4 = 13 \end{cases}$$

Agora basta calcular os valores de $x_1; x_2; x_3$ e x_4.

Sistemas Lineares

Fase da substituição

Substituindo uma equação na outra...

Sistema reescrito

Equação (4): $13x_4 = 13 \Rightarrow x_4 = 1$

Equação (3): $3x_3 + 13 = 13 \Rightarrow x_3 = 0$

Equação (2): $x_2 - x_3 - 5x_4 = -7 \Rightarrow x_2 = 0$

Equação (1): $x_1 + x_2 + 3x_4 = 4 \Rightarrow x_1 = 1$

Sistemas Lineares

Exercícios:

$$\begin{cases} 3x_1 + 2x_2 + 4x_3 = 1 \\ x_1 + x_2 + 2x_3 = 2 \\ 4x_1 + 3x_2 + 2x_3 = 3 \end{cases}$$

Encontro 11: Sistemas Lineares + Aplicações

Aplicações

Enunciado do problema.

- Passo 1
- Passo 2

O objetivo desta aula é resolver problemas práticos com programação.

1. Abra o Python
2. Siga as instruções

Encontro 12: Sistemas Lineares

Objetivos

Critério de parada

Gauss-Jacobi

Gauss-Seidel

Critério de parada

Podemos usar o critério de parada/convergência para ambos os métodos de resolução de sistemas lineares. Os sistemas iterativos deverão ter um critério de parada.

Epsilon

$$= \frac{maxkx^{k+1} \quad x^{(k)}k}{maxkx^{(k+1)}k}$$

Gauss-Jacobi

É baseado na decomposição da matriz A em uma soma de três matrizes D, L e U, onde D é uma matriz diagonal contendo os elementos diagonais de A, L é uma matriz triangular inferior com os elementos fora da diagonal de A, e U é uma matriz triangular superior com os elementos fora da diagonal de A.

Gauss-Jacobi

$$x^{(k+1)} = D^{-1}(b - (L + U)x^{(k)})$$

onde $x^{(k)}$ é o vetor de aproximação na iteração k, e D^{-1} é a inversa da matriz diagonal D.

Gauss-Seidel

Assim como o método de Gauss-Jacobi, ele também é baseado na decomposição da matriz A em três matrizes: D, L, e U. No entanto, o método de Gauss-Seidel atualiza as aproximações das incógnitas de forma sequencial, usando os valores mais recentes disponíveis.

Gauss-Seidel

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1} a_{ij} x_j^{(k)} \right)$$

onde $x_i^{(k)}$ é a aproximação da incógnita x_i na iteração k, a_{ij} é o elemento na linha i, coluna j da matriz A, e b_i é o i-ésimo componente do vetor b.

Gauss-Jacobi-Seidel

Resolva esse sistema linear utilizando ambos os métodos e considere o erro em 0,1. Responda com quantas iterações ambos os sistemas convergiram.

Sistema linear

$$\begin{cases} 10x_1 + 2x_2 + x_3 = 7 \\ x_1 + 5x_2 + x_3 = 8 \\ 2x_1 + 3x_1 + 10x_3 = 6 \end{cases}$$

Importante!

Este material é exclusivo de uso do autor. Proibido copiar ou replicar.

rogeriovargas@ufpr.br

GEOIDELAB

UFPR

Cálculo Numérico

Um guia prático com Python

Prof. Dr. Rogério Vargas[1]

[1]Centro de Estudos do Mar
Universidade Federal do Paraná

2024

GEOIDELAB

UFPR